## GwentCard游戏设计

基于学习的目的，仿照巫师3狂猎中的昆特牌做了一个小的Demo，基本实现了游戏机制和卡牌效果。因为是初学的缘故，还有很多小的Bug，但不影响整体，游戏收录了巫师3的开场动画和3首BGM。卡牌包含了除了史凯利特群岛卡组之外所有的昆特牌。

# 使用游戏技术

1. Unity3d引擎
2. NGUI界面插件

游戏的2D界面，背景，按钮，文字，动画等均使用NGUI插件功能实现，后面做详细解析。

1. Prefab预制体

常用的需要作统一保存的部件使用预制体处理，游戏中每个单独的部件在基础上再做细微的调整，如，每张卡牌，大部分功能相同，但是卡片的图片和效果不同。

1. Atlas图集

游戏中图片调用到NGUI插件时，需要使用Atlas图集作保存，才能使用。

# **每个场景的功能**

1. Main场景

进入游戏界面，4个按钮，分别是开始游戏，配置卡牌，开场动画和退出游戏。

1. Configuration场景

配置卡牌场景，最上面分别是4个卡组。点击卡组后，可在左上角下拉框里选择领导牌、特殊牌、生物牌、中立牌4中卡牌类型。其中，领导牌只能有一张，特殊牌最多10张，生物牌加中立牌最少25张，最多40张。点击选择可选卡牌，点击+和-可设置同一卡牌的数量，左下角是当前卡组的卡牌数量，下方是当前卡组的卡组效果，右下角是保存和返回按钮。

1. Play场景

游戏主场景，进入游戏后，最开始会选择卡组，如果没有配置卡牌那么卡组就不可选择。选择卡组后，开始换牌，可换两张牌。然后开始游戏。游戏开始，会自动计算先手后手，然后正常游戏，三局两胜。游戏棋盘分为己方和敌方，分别都有领导牌，姓名版，手牌，坟墓，卡牌库，近战、远程、攻城区域，区域右上角有眼睛标志，可放大显示卡牌，而且手牌区域点击眼睛可出牌。近战、远程、攻城区域有战斗号角区域、天气可覆盖、左边可计算战力。姓名版可显示手牌数量，大比分，和总战力。领导牌区域如果效果可用则显示红色，否则显示蓝色。游戏进行过程中会有出牌和游戏信息动画，右方有结束当前回合和设置按钮。设置按钮可关闭背景音乐，重新开始或者返回主菜单。游戏结束后会计算游戏结果。

# **资源文件夹**

1. Atlas游戏图集
2. Musics 存放BGM和开场动画
3. NGUI界面插件
4. Prefabs游戏预制件
5. Scenes场景
6. Scripts游戏脚本
7. Textures存放图片资源

# **游戏脚本解析**

1. **Global.cs**

需要用到的全局变量

* 1. 枚举

Group 卡组类型

List 卡牌类型

Line 卡牌的排

Effect 卡牌的效果

* 1. 常量

Path 玩家配置的卡组保存地址

EnemyPath 敌人的卡组保存地址

* 1. 方法

SetTarget 将手牌打出到指定的出牌区域

GetItemInt 获取列表的序数

1. **CardProperty.cs**

卡牌需要的变量

Line 卡牌的排

Effect 卡牌的效果

Glod 是否为金色卡牌

Power 卡牌是战力

1. **LoadScene.cs**

加载场景

* 1. 序列化变量

Scene 目标场景

* 1. 方法

OnClick 点击按钮

1. **MusicManager**

音乐管理器

* 1. 序列化变量

AudioClip 音乐数组

AudioSource Unity音频部件

* 1. 变量

Pause 是否暂停

* 1. 方法

Awake 初始化

FixedUpdate 每帧运行，判断可否播放下一首BGM

Pause 暂停音乐

Continue 继续播放音乐

1. **Singleton.cs**

单例模式

* 1. 静态变量

Instance 单例实体类

* 1. 静态方法

GetInstance 获取单例实体

1. **CoroutineManager.cs**

队列管理器，用于逐个播放动画，不出现冲突

* 1. 变量

Queue 队列

Current 队列当前项

IsFinish 队列播放是否结束

* 1. 方法

FixedUpdate 每帧运行，DoTask方法

AddTask 添加项目到队列中

DoTask 队列管理主方法，逐个运行队列中的项目

Finish 结束队列播放

GetFinish 返回是否结束队列布尔值

1. **Main/Exit.cs**

退出游戏

方法 OnClick 点击按钮退出游戏

1. **Main/Movie.cs**

播放过场动画

* 1. 序列化变量

VideoPlayer Unity视频部件

* 1. 方法

Awake 初始化

OnGUI 界面绘制界面方法，用于点击视频后退出播放

OnClick 点击按钮开始播放

1. **Confuguration/CardPlus.cs**

选择或增减卡牌

* 1. 序列化变量

Label 卡牌数量标签

Toggle 选择卡牌单选框

* 1. 变量

Total 当前卡牌数量

Max 最大卡牌数量

* 1. 方法

Awake 初始化

Plus 增加卡牌数量

Minus 减少卡牌数量

WriteTotal 显示当前卡牌数量

Check 选择卡牌

1. **Confuguration/EnemySaveController.cs**

自动配置敌方卡组

方法 Start 进入场景时运行，自动配置敌方卡组

1. **Confuguration/LeaderCardCheck.cs**

选择领导牌

方法 Check 选择领导牌

1. **Confuguration/NumberController.cs**
   1. 序列化变量

Labels 总卡牌数标签数组

* 1. 变量

LeaderCount 领导牌数量

SpecialCount 特殊牌数量

MonsterCount 怪物牌数量

* 1. 方法

Number 计算总卡牌数并显示

1. **Confuguration/SaveController.cs**

保存卡组

* 1. 序列化变量

Label 提示信息标签

ScrollView 卡牌区域

Xml 保存文件类

* 1. 方法

Start 进入场景时运行，加载已保存的卡组配置

Initialize 如果没有配置，则初始化

LoadXML 从文件中加载卡组配置

UpdateXML 更新卡组配置

OnClick 点击按钮保存卡组配置到文件

ShowLabel 显示提示信息

* 1. 异常类

SaveException 自定义异常类，内容为空，只用于当前脚本

1. **Confuguration/TagController.cs**

选择卡牌类型并显示

* 1. 序列化变量

Groups 选择卡组区域ScrollView

TagButtons 卡组类型GameObject

PopupList 下拉列表

GroupLabel 卡组效果Label

Grids 选择卡组区域

* 1. 变量

Group 当前卡组类型

List 当前卡牌类型

* 1. 常量

GroupLabelText 卡组效果文字

* 1. 方法

Start 进入场景时运行，显示默认卡牌

OnClick 点击卡组类型按钮时显示

OnValueChange 点击卡牌类型下拉框时显示

GetGrids 获取当前卡组类型的卡组区域

1. **Play/BlackShow.cs**

显示或者隐藏黑色遮罩层

方法 Show 显示或者隐藏黑色遮罩层

1. **Play/ChooseGroup.cs**

游戏开始时选择卡组

* 1. 方法

Start 进入场景时运行，获取卡组信息

OnClick 选择点击卡组后开始游戏

1. **Play/CardBehavior.cs**

当前卡牌的出牌行为

* 1. 变量

CardProperty 卡牌需要的变量

TotalPower 卡牌的当前战力

* 1. 方法

Awake 初始化

Play 打出当前卡牌，根据当前卡牌的变量表现不同的行为

Dummy 如果当前卡牌是假人，则执行此行为

Replace 替换卡牌阶段执行此行为

1. **Play/GameController.cs**

游戏主逻辑

* 1. 序列化变量

Atlas 图片集

CardPrefab 卡牌预制体

player\_life\_gem 玩家生命宝石

enemy\_life\_gem 敌方生命宝石

* 1. 枚举

GameBehavior 当前局输赢

* 1. 变量

Offensive 先后手

player\_fail 玩家输的局数

enemy\_fail 敌方输的局数

* 1. 方法

Initialize 初始化，加载玩家和敌方手牌并开始换牌阶段

StartGame 开始游戏，决定先后手并发动领导牌技能

Turn 结束一局，开始下一局，切换先后手，清场，判断是否游戏结束等

1. **Play/PlayerController.cs**

玩家类

* 1. 序列化变量

Grids 玩家手牌，出牌等区域

Obj 玩家父节点

avatar\_group 玩家卡组类型图片

group\_label 玩家卡组类型标签

deck\_realms 玩家牌库图片

number\_label 玩家手牌数标签

deck\_realms\_label 玩家牌库数标签

* 1. 变量

Group 玩家卡组类型

* 1. 方法

Initialize 初始化

DrawCards 出牌或者将牌放在指定区域

Number 计算战力和手牌数

PlayOver 结束当前回合

1. **Play/EnemyController.cs**

**敌方类**

* 1. 序列化变量

Grids 敌方手牌，出牌等区域

Obj 敌方父节点

avatar\_group 敌方卡组类型图片

group\_label 敌方卡组类型标签

deck\_realms 敌方牌库图片

number\_label 敌方手牌数标签

deck\_realms\_label 敌方牌库数标签

* 1. 变量

Group 玩家卡组类型

* 1. 方法

Initialize 初始化

DrawCards 出牌或者将牌放在指定区域

Number 计算战力和手牌数

PlayOver 结束当前回合

Play 敌方进行游戏方法

1. **Play/AIController.cs**

敌方AI行为类

* 1. 变量

CardList敌方手牌列表

* 1. 方法

AICard AI出牌逻辑

AITurn AI结束回合逻辑

HasCard判断是否有指定类型的卡牌

IsWeather判断是否有当前天气效果

1. **Play/PowerController.cs**

计算战力类

* 1. 序列化变量

player\_power\_label 玩家战力标签数组

enemy\_power\_label 敌方战力标签数组

player\_winner\_indicator 玩家赢家指示器

enemy\_winner\_indicator 敌方赢家指示器

* 1. 变量

player\_total 玩家战力总数

enemy\_total 敌方战力总数

Player 玩家战力数组

Enemy 敌方战力数组

* 1. 方法

Number 计算战力方法

1. **Play/TurnController.cs**

回合类

* 1. 序列化变量

PlayerLabel玩家回合结束标签

EnemyLabel敌方回合结束标签

TurnButton回合结束按钮

* 1. 变量

TurnIndex回合数

IsTurned是否已结束回合变量

* 1. 方法

FixedUpdate 每帧运行，判断回合结束按钮是否可用

PlayerTurn玩家结束回合

EnemyTurn 敌方结束回合

Clear下一回合

1. **Play/ShowCards.cs**

显示卡牌类，出牌功能就在这里实现

* 1. 枚举

ShowBehavior 显示卡牌行为类型

* 1. 序列化变量

Grid 显示卡牌区域

PopupList选择排下拉列表

Obj 显示卡牌父节点

Label 标题标签

MessageLabel提示信息标签

ScrollView 滚动区域

OKButton 确定按钮

ReturnButton 返回按钮

* 1. 变量

Card 选择的卡牌

TotalGrid 当前卡牌区域

TotalLine 当前排

ReplaceInt 替换卡牌的数量

* 1. 方法

Show 显示卡牌

ShowLeader 显示领导牌

Hide 隐藏显示

LineChanged 下拉列表切换排后调用

1. **Play/ShowButton.cs**

显示卡牌按钮

* 1. 序列化变量

Behavior 显示卡牌时的行为

Grid 显示卡牌的目标区域

* 1. 方法

OnClick 点击眼睛，显示卡牌

1. **Play/SetController.cs**

设置界面

* 1. 序列化变量

Obj 设置父节点

Toggle 背景音乐选择框

Button 返回按钮

* 1. 方法

Show 显示设置界面

Check 开关背景音乐

Return 返回游戏界面

1. **Play/GameOver.cs**

游戏结束界面

* 1. 序列化变量

Obj 游戏结束界面父节点

Label 标题标签

player\_labels 玩家标签数组

enemy\_labels 敌方标签数组

* 1. 变量

PlayerPowerList玩家战力数组

EnemyPowerList敌方战力数组

* 1. 方法

Show 显示游戏结束界面

AddPower 回合结束后，添加当前战力到数组

1. **Play/WarhornController.cs**

战斗号角类

* 1. 序列化变量

PlayerGrids 玩家战斗号角区域

EnemyGrids 敌方战斗号角区域

* 1. 变量

PlayerWarhorn 玩家对应排战斗号角

EnemyWarhorn 敌方对应排战斗号角

* 1. 方法

Warhorn 打出战斗号角牌的行为

1. **Play/AgileController.cs**

敏捷卡牌类，多用于松鼠党牌组

方法 Aglie 打出敏捷牌的行为

1. **Play/ScoiataelController.cs**

松鼠党类，松鼠党的能力是可选择先后手

* 1. 序列化变量

Button选择先后手按钮

PopupList 先后手列表

* 1. 方法

Start 进入场景时运行，判断按钮是否可用

Awake 初始化

OnValueChange 选择先后手后运行

1. **Play/WeatherController.cs**

天气类

* 1. 序列化变量

Grid 天气出牌区域

PlayerFrostSprite玩家霜图片

PlayerFogSprite玩家雾图片

PlayerRainSprite玩家雨图片

EnemyFrostSprite敌方霜图片

EnemyFogSprite敌方雾图片

EnemyRainSprite敌方雨图片

* 1. 变量

Weather对应排天气变量

* 1. 方法

ClearSky 晴天

Frost霜

Fog雾

Rain雨

1. **Play/MusterController.cs**

召集卡牌效果，多用于怪兽牌组

* 1. 常量

MusterCards 可召集的卡牌组合

* 1. 方法

Muster打出可召集怪物的行为

1. **Play/LeaderController.cs**

领导牌类

* 1. 序列化变量

Obj领导牌父节点

turn\_indicator 领导牌指示图片

* 1. 常量

turn\_indicator\_string 领导牌指示名

* 1. 方法

PlayerTurnIndicator 玩家领导牌指示行为

EnemyTurnIndicator 敌方领导牌指示行为

1. **Play/TweenCard.cs**

打出卡牌动画效果

* 1. 序列化变量

Obj 动画效果父节点

* 1. 方法

Play 该动画的行为

1. **Play/TweenMessage.cs**

提示信息动画效果

* 1. 序列化变量

Obj 动画效果父节点

Label 提示信息标签

* 1. 方法

Play 该动画的行为

# 领导牌实现方式

领导牌使用了面向对象的继承，首先是领导牌基类，然后继承为玩家和敌人两个子类，然后玩家和敌人各实现自己的领导牌子类，通过重重继承的方式实现了领导牌的效果，精简了代码。

1. **代码中不易理解的地方**
2. 单例模式：由于每个脚本在场景中只有一个，但是调用时不能使用new对象的方式，所以可以使用单例模式取得对应脚本的方法，如：‘BlackShow.GetInstance().Show(false);’通过GetInstance就取得了BlackShow对象，并可使用其中的方法，当然，也可使用 Unity自带的FindObjectOfType取得对象，效果都是一样的。
3. 协程：Unity自带了协程模式，再通过自己设计的队列管理器，就实现了动画一个一个的播放。协程方法返回值类型必须为IEnumerator，然后通过yield return实现动画暂停等功能，调用方法为StartCoroutine，具体请参考源码。
4. Lambda和虚函数：代码中使用到委托时，除了可通过NGUI插件自带的界面拖拽外，代码中也会使用到，具体代码为EventDelegate.Add(event,callback)，其中，event是对应事件，callback是对应方法，callback可用lambda和虚函数表示，lambda的写法为‘()=>+一行代码’，当需要使用到多行代码时，可使用虚函数，写法为‘delegate{多行代码;}’。